



Daffodil
International
University

Department of CIS

Summer 2021

Course: Internship

Course code: CIS499

Project Title: API connecting from web based application to desktop app of “HEXAM” and cost effective lms using google classroom.

Supervised by:

Nayeema Rahman

Sr. Lecturer

Submitted by:

Walid Hasan

(183-16-349)

Submission Date: 22/11/2021

APPROVAL

Project title” API connecting from web based application to desktop app of “HEXAM” and cost effective lms using google classroom”, submitted by Walid Hasan, ID No: 183-16-349 to the Department of Computing & Information Systems, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing & Information Systems and approved as to its style and contents. The presentation has been held on 02-11-2021.

BOARD OF EXAMINERS



Mr. Md Sarwar Hossain Mollah

Chairman

Assistant Professor and Head

Department of Computing & Information Systems

Faculty of Science & Information Technology

Daffodil International University



Ms. Nayeema Rahman

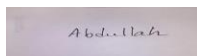
Internal Examiner

Sr. Lecturer

Department of Computing & Information Systems

Faculty of Science & Information Technology

Daffodil International University



Mr. Abdullah Bin Kasem Bhuiyan

Lecturer

Department of Computing & Information Systems

Faculty of Science & Information Technology

Daffodil International University

Internal Examiner

Dr. Saifuddin Md. Tareeq

Professor

Department of Computer Science and Engineering

Dhaka University, Dhaka

External Examiner

Declaration:

I hereby declare that; this project has been done by me under supervision of Ms. Nayeema Rahman, Senior Lecturer, department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of there has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

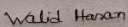
Supervised By

Ms. Nayeema Rahman

Sr. Lecturer

Department of CIS

Daffodil International University

Submitted By

Walid Hasan

ID: 183-16-349

Department of CIS

Daffodil International University

Executive Summary: This is an academic project. Here my work is to develop API that will be integrate web application and desktop app HEXAM. As a result the proctoring system will be enable. Our desktop app name is HEXAM. This project is safe. Using this system student can give exam online. And their assessment will be easy. This system ensures safety for the users and this project will increase Government revenue.

Acknowledgement: First of all I will like to thanks to Almighty Allah for keeping my health good so I am able to do this project and documentation. I will also humble to **Mr. Mazharul Islam** sir for continuous support and also I will like to cordially thanks to my supervisor **Ms. Nayeema Rahman** for giving proper instruction. At last my teachers, team members, family and friends for always supporting and inspired me doing this project. My parents and teachers always supports me. Without this support doing this project was quite tough.

Abstract: My project name is proctoring that I am working. There is a web application and desktop app name HEXAM. It's an online exam taken system, in here AI is enable. So there is no option for cheating. For using this system a person need to have a smart device. Like mobile, pc or laptop. In between my research there are a lot of system is available in the market. But those system aren't free. Our Proctor system will be free, and it will monitor the exam. In google classroom there will be also option for taking exam by google form. This type of system is not free, and we are trying to bring it to the market.

Contents

Executive Summary	v
Acknowledgement:	v
Abstract:	v
Chapter 1- Introduction	1
Chapter 2- Initial Study	1
Background of the project.....	1
Problem area:	2
Possible solution.....	2
Main Function:	2
Chapter 3-Literature Review	3
Comparison of leading solutions	3
Best features	4
Limitation:	4
Chapter 4- Planning	4
Project Plan	4
Work Breakdown structure:.....	4
Gantt chart:.....	6
Chapter 5- Feasibility Analysis.....	6
Cost Benefit Analysis	7
Agile Software Development	7
Chapter 6- Foundation.....	8
Overall requirements List:	8
Non-functional requirement	8
Technology.....	8
Chapter 7-Exploration	9
Architectural Design.....	9
Use case:	9
Full system Activity Diagram:	11
Class 8- Engineering	14
Class Diagram:	14
ERD Diagram:	15

Sequence Diagram:.....	16
Chapter 9- Development	17
Core modules coding samples.....	17
Possible problem breakdown	20
Chapter 10-Testing.....	20
Chapter 11- Implementation.....	24
Training.....	24
Chapter 12- Critical Appraisal and Evaluation.....	25
Success rate against each objectives.....	25
How much better could have been done	25
How better the feature of the solution	25
Which features could not be touched	25
Chapter 13- Conclusion.....	26
Summary of the project.....	26
Goal of the project	26
My experience	26
Appendices	27
Test Scripts.....	27
System code	28
References	31
Plagiarism Report.....	31

Chapter 1- Introduction

Every project has some specific features and aims. That the project wants to achieve. Every project has one common aim that is to solve problems. In our country there is no such system like proctoring and that is free of cost for use. This types of system has increased its demand huge during this pandemic. Because online class was ongoing but there was no such system for taking exam. So students falls behind and there is a chance for study gap and year loss.

Here I am working on project name is Proctoring and cost effective lms using google classroom. My task is to develop API for desktop app HEXAM. The system will have two type of user mainly. One will be Student and other one will be teacher. Teacher will set question and student will give exam. This system will be a web application. And the main proctoring system name is HEXAM. My main task is to connect the both system that's why this system can do data transfer. And that's why the proctoring system s will be enable.

And in the google classroom there will be option available for giving exam.

Chapter 2- Initial Study

During this lockdown due to pandemic whole world faces a lot of crisis. The world goes under shut down situation. All the things was closed. School, collage, University also. Although there was online classes but unfortunately there was no such great way for giving exam. And also for proper assessment. It's a big problem. Private University somehow build system like proctoring and can hire. So they can manage. But unfortunate for National University and some university that have limited budget falls behind. Result study gap, year loss, lose interest for study and depression. There are some system available like proctoring. But it was not free and there are lot of extra function. Here comes our idea with a positive aspect. Here we are making web application. And a desktop app name HEXAM. In the desktop app AI is enable. I connect these two system for data transfer. We will make cost effective lms that will be attach with google classroom. Google giving a lot of facility without giving exam. Google classroom is a common platform and very popular also. If we can create the exam option here, it will be cost free and there will be no server issue. Free of cost a lot of school, and collage can use this for taking exam and also for giving assessment. Our main purpose is to make a system that is free of cost and easy to use and manage. That's why this idea was build up and I am working on this project.

Background of the project:

In our country there is no such centralized system to take exam online. Online class somehow can be managed. Other developed country there is available system like proctoring. During any kind of shut down situation, pandemic they can use it. But unfortunately there is no system in our country.

So our main project name is Proctoring. One will be web application. Where the question will come. HEXAM is a desktop app. In here AI is enable. When the exam will start the webcam will

be open. It will monitor user activity. Any suspicious activity if happen like cheating, extra tab opening, and eye movement all will be noticed. And there is also other functions. And after finishing exam system will assess over this. Online exam and monitoring is happening is main priority.

This project is based on online exam taking side. Here our system web application will be integrated with google classroom. Google classroom is very familiar with the students and teacher. Everybody now days are using it, for classroom and submission something like assignment. The idea that we are working is to connect google classroom. Here our team have built a web application that will be integrated with google classroom. This type of project on my perspective have not built yet. It's a new and unique idea. Before google classroom there was only the classroom option just and when the system will be okay there will be exam option available for use.

Problem area:

This solution actually solves a lot of problem. There are lot of proctoring system that is available in the market. These solution is not free. Need a certain amount of charge and also free subscription time is limited. During this pandemic and also other time this type of software has great demand. Our desktop application will be free of cost. And from the web application connecting google classroom is making. So google classroom ability will be increase. That will not be just google classroom. There will be also available option for giving exam.

Possible solution: This system will be just web application that will be connected with desktop app HEXAM with rest API. And google classroom will be connected by google app script. And there will be no mobile app service.

Main Function: There are a lot of function available. I am giving a list of core function that the system will perform sure.

- HEXAM will be integrated to web application as a result resource can be shared.
- And the desktop app there is main proctoring features that will be enable.

- Giving exam in desktop app, user needs information, which will come through API.
- API allows to search data by certain criteria by your or by ones choice.
- In the google classroom there will be options for giving exam in google form.

Chapter 3-Literature Review

Comparison of leading solutions: Proctoring helps as an online supervisor. When there are different types of online exam taken application. One name is **exam online**. This app has some best features. It has AI enable system that has authentic authorization identifying face issue, multiple face present or not, movement detection of facial, warning, different marking scheme, window cannot be changed etc. (ExamOnline, n.d.).

Home Page:

- Open the URL provided for Exam in Chrome Browser
- Try out with <https://assessment.examonline.in/MockTest/Exam/> (Mock)
- You will see the Instructions page
- Please read the instructions carefully
- Ensure all set up is done as per the instructions
- Click on 'Continue'



EXAMONLINE
Reaching out to everyone...

EXAMONLINE
Reaching out to everyone...

Kindly Read the following instructions

CAUTION:

1. Candidates must note that though the test is being taken at their place of stay yet their conduct during examination will be proctored using Artificial Intelligence as well a Human proctor.
2. Candidate's real time video and audio feed is being transmitted to the Proctor for invigilation.
3. Candidate should note that examination authority have zero tolerance policy for unethical behaviour.
4. It is strongly advised that candidates take the examination with utmost solemnity and ensure that all the instructions are strictly followed.
5. Exam Authority reserves the right to disqualify any candidate who is found to be indulging in any kind of malpractice.

Continue

SoftwareSuggest.com

Best features: After connecting API from the web application to the desktop application all the data can be transfer. So exam can be given. And the main proctoring features will be shown. In our system AI is enable. User authentication, if there multiple face present or not, warning, eye contact etc.

Limitation: Connecting API, data is passing, but we need to test the system more and more for publication. No offline recording system. It's just work online.

There are google classroom that is mostly popular. Here in google classroom there is only option for view something and to submit or publish something. Here my work is to develop API that will be integrated with google classroom through google app script. This platform is just for education and in here the exam option is not available.

Best features: Google classroom will connect to the web application database through google app script. After run it will go to google sheet. From the google sheet google form will generate. And it will send to google classroom. That's how it works.

Limitations: The system of publication need more testing. Here we can give exam by google form.

Chapter 4- Planning

Project Plan: In this part I am going to show how I have organized my work. How the full work is going to divided into small parts and the needs to be completed in fixed time. Here it is.

Work Breakdown structure: Full project work is huge. Needs to be breakdown in small parts. There is a lot of task. Needs to divide this task in small parts so that I can meet the deadline. It will be easier and more efficient. I have followed this breakdown structure. Without this, completing the whole task was tough.

	A	B	C	D
1	Task Name	Duration	Start	Finish
2	Introduction	3 days	9/8/2021	9/10/2021
3	Initial Study	3 days	9/11/2021	9/13/2021
4	Literature Review	4 days	9/14/2021	9/17/2021
5	Planning	6 days	9/18/2021	9/23/2021
6	Feasibility	12 days	9/24/2021	10/4/2021
7	Foundation	4 days	10/5/2021	10/8/2021
8	Exploration	5 days	10/9/2021	10/13/2021
9	Engineering	8 days	10/14/2021	10/21/2021
10	Deployment	3 days	10/22/2021	10/24/2021
11	Testing	2 days	10/25/2021	10/26/2021
12	Implementation	1 days	10/27/2021	10/27/2021
13	Critical Appraisal and Evaluation	4 days	10/28/2021	10/31/2021
14	Conclusion	1 days	11/1/2021	11/1/2021
15	Total	56 days		

Gantt chart:



Chapter 5- Feasibility Analysis:

Feasibility analysis is very important for a project. It will shows how this system will help in amount of work. How it will help an organization and will reduce the work pressure. How this system will be helpful. Feasibility study will evaluate a systems main potential. It shows description of the project. Every time new plan is coming, but when we think to implement it, it must go under feasibility analysis. A group discussion must needed for feasibility analysis. For connecting one system with other system we must need API integration.

- **Technical Feasibility:** This shows in which technology I am going to build this API that will be integrated. Does this technology needs deployment? Making this project in this technology is easy or not. Here we are using mainly python and its framework Django. API design I have used rest API Django. This is quite popular all over the world. And its

implementation is also easy. Technical feasibility also check does the user of system can easily use it or not. Now a day's people are smarter and can easily use this system.

- **Operational Feasibility:** This gives us the ability if the new system solves all the problems that the existing system have. If I build API once. That can be used again and again to connect other system. So we can use system as much as we want and to connect. Once we develop API it can be used anytime.

Cost Benefit Analysis: It is a very important part for a project. There are a lot of API available. Does building this API will cost more, how will it solve problems. This analysis will give answer this question. This analysis will give a clear cut answer of this.

Time	Days	Money
Researching Time	2	
MVP Time	2	
Developing API time with database design	5	
Integration time with main system	2	
Maintenance	2	
Documentation	3	
		Total cost=20000

API cost is not much. So it has economic feasibility to use.

Agile Software Development: In software development model list there is a lot model available. In my project I am going to agile software development model. This development model is flexible. Work can be divided. Customer satisfaction is a big part in this development model. Changing requirements, early delivery. Working software and sustainable development. Focus on the quality development. Taking feedback from the customer continuously. These requirements matches my development model. Here in our system our work is divided. Everyone is busy on everyone individual work. So quality development is maintained. I am taking feedback about my work continuously. So I am ensuring sustainable development. That's why agile development is perfect and for my work also. (Agile Software Development, n.d.)

Chapter 6- Foundation

Overall requirements List: These are the functional requirement. Functional requirement shows what a system do.

No	Requirement list	Moscow Priority
FR-1	API will connect from web application to desktop app HEXAM.	Must have.
FR-2	API will connect the database.	Must have.
FR-3	Data will be transferred.	Must have.
FR-4	Security issue.	Must have.
FR-6	All the information will store in database.	Must have.
FR-7	Confirmation mail.	Should have.
FR-8	Google app script will link to web application database.	Must have.
FR-9	Google form will be generated.	Must have.
FR-10	The system will be able to give exam.	Must have.

Non-functional requirement: These are mainly a systems capabilities. It's a systems enhance functionalities. Example:

- Speed: After connecting API how fast the system response. Speed is a major issue for the user. If the speed is not good, user will not like it and lose their interest.
- Security: It's a major concern issue for this time. So all the important information needs to handle. Before connecting other system thorough API we need to judge properly.
- Scalable: When work load is much API is scalable or not.
- Usability: How easy one can use the system. Is it easy to use? User can understand all the functionality easily or not. So for this we need proper user feedback. (Non functional requirement, n.d.)

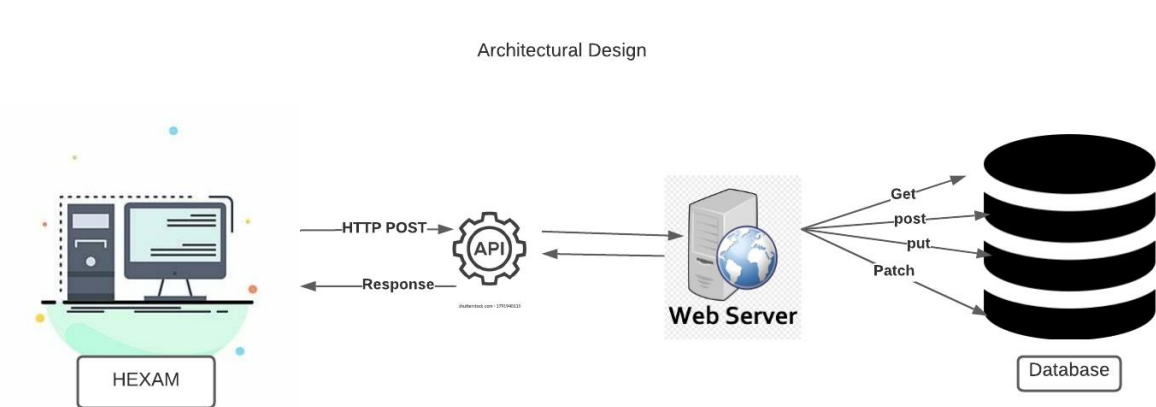
Technology: First there will be a web application. And the main proctoring system name will be 'HEXAM' that will be a desktop app.

In here python is used. Python is widely used all over the world. Python ranked second among all the programming language. It's popular because its syntax is easy, and needs to write programming language less and small. It has huge library. Python is a high level programming language, and

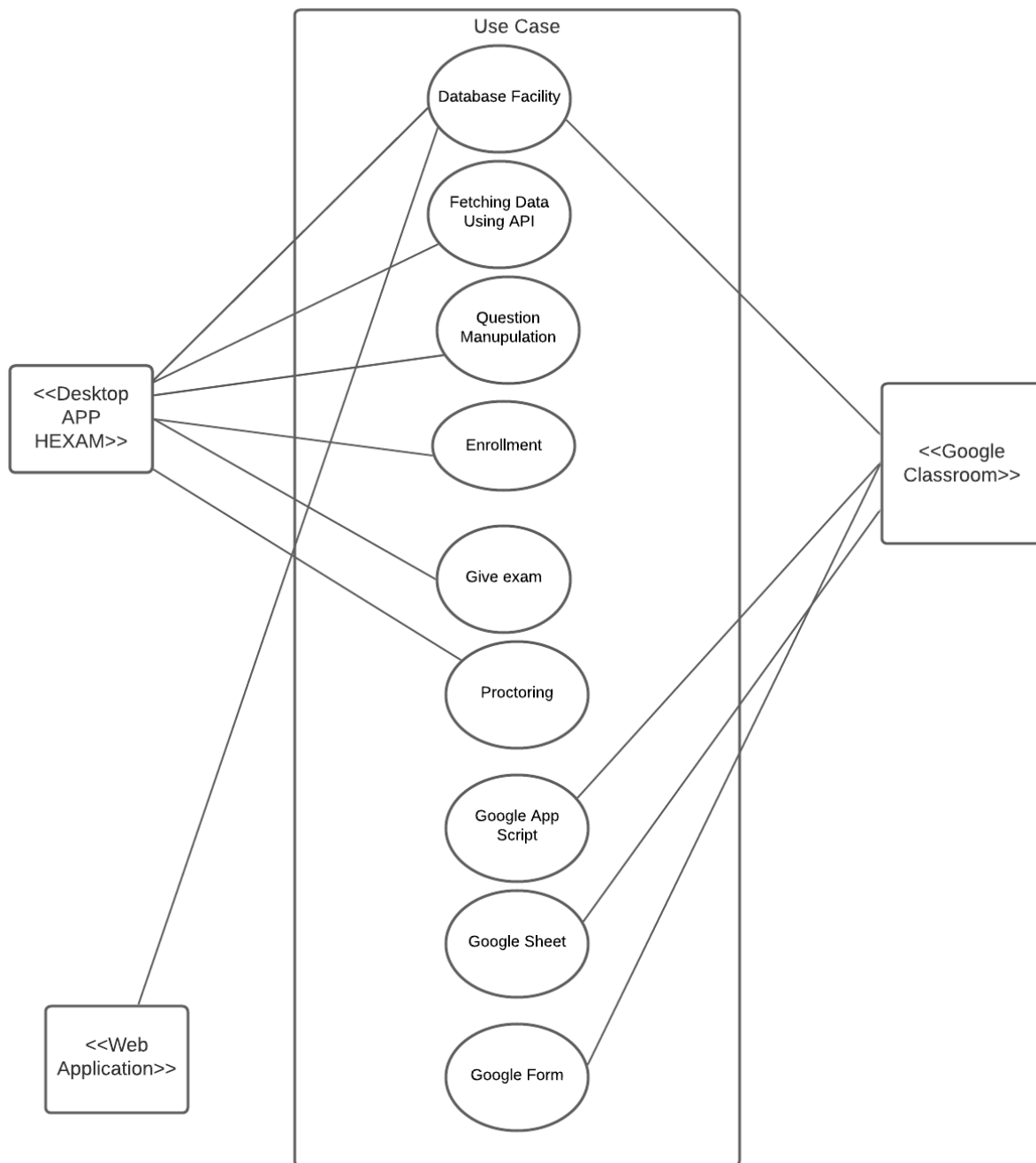
because of its object oriented approach can develop quickly and can write clearly. In here the framework Django has been used. I have used rest API Django to design API.

Chapter 7-Exploration

Architectural Design: This design shows the combination of hardware and software of a system.



Use case: Use case is very important part for understanding a system. It describes how the system will interact.



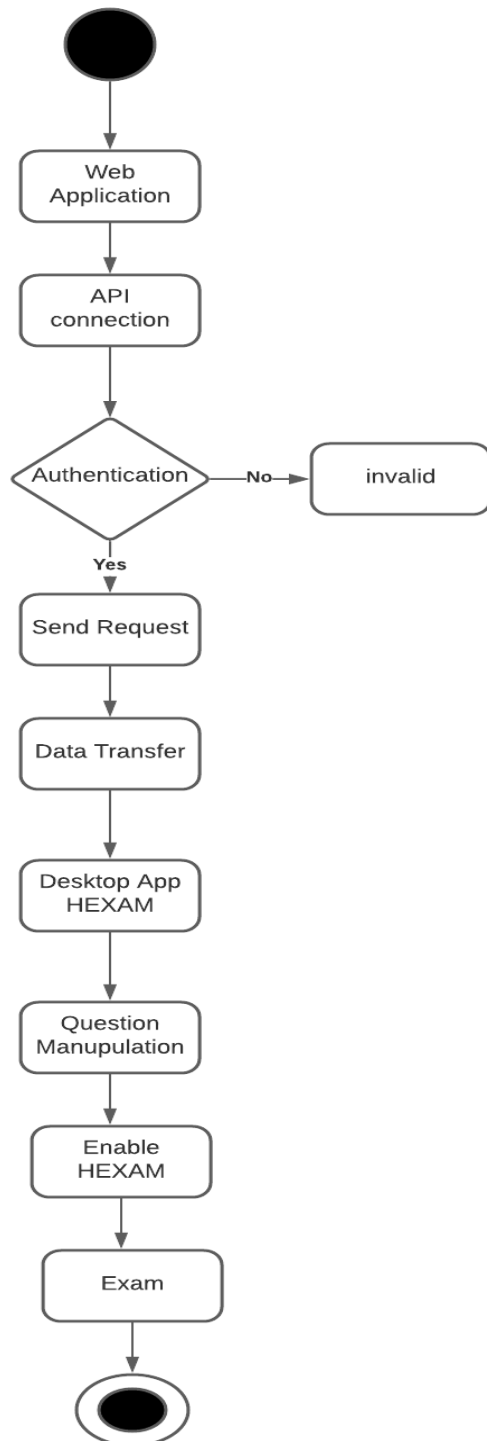
In here HEXAM is interacting with web application database. Fetching data using API. Question Manipulation is happening. Enrollment process will show. Student can attempt exam. Proctoring will start in.

Full system Activity Diagram:

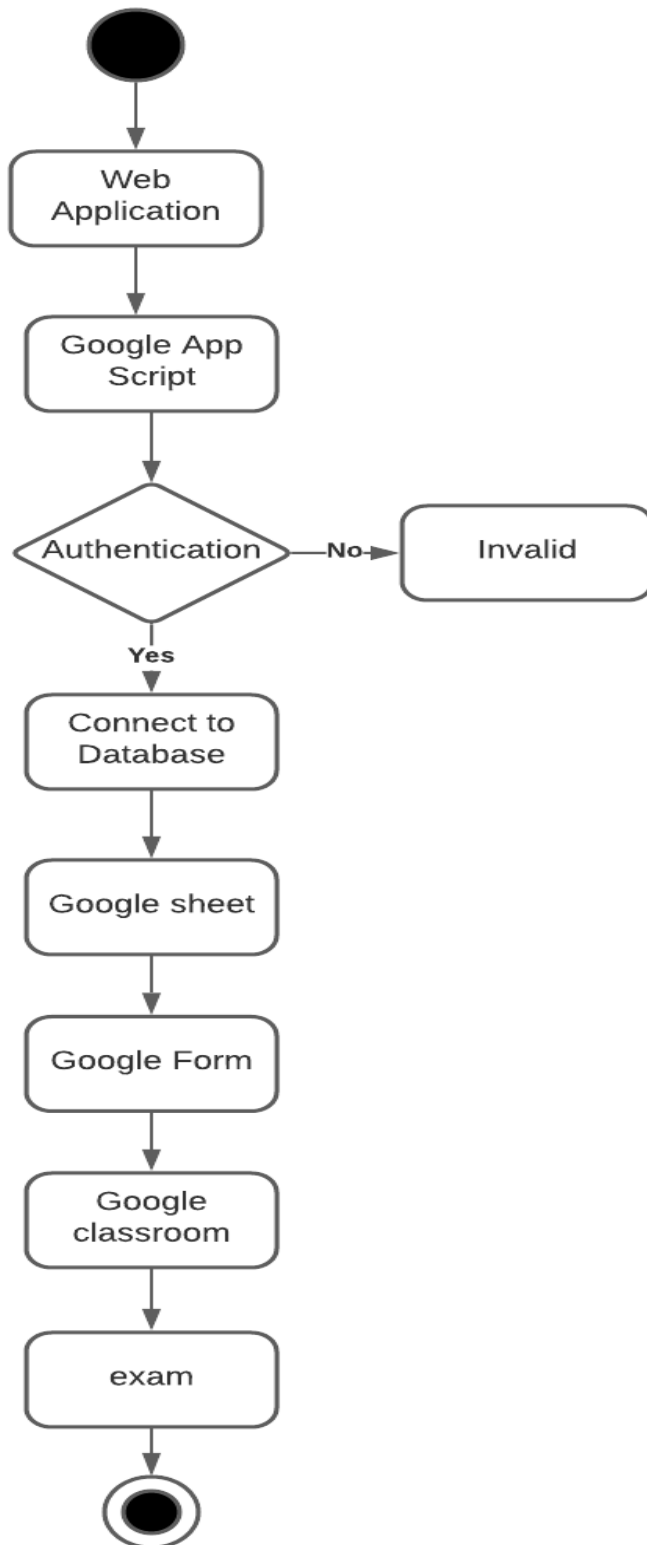
This diagram is a behavioral diagram mainly. It focuses on the behavior of a system. It also shows starting and end point of a project.

Desktop application name is HEXAM. It will be connected with web application through rest API. If only authentication then can send request. And data connection can happen then to the desktop app HEXAM. Question manipulation will be start. Proctoring will start. Can give exam. Finish.

Activity Diagram Of
HEXAM



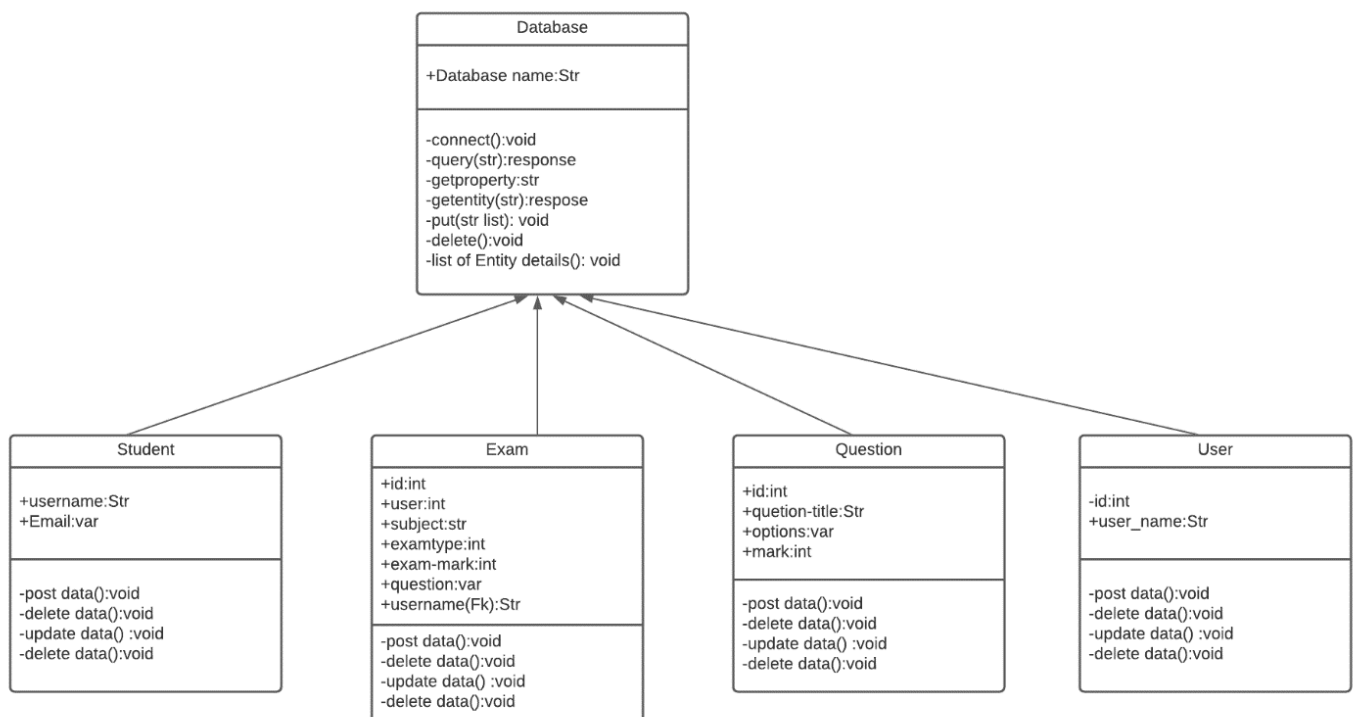
Activity Diagram For Google Classroom



Here I connect the web application to google app script. From the web application the question will be generated. When I run the app script the question will go to google sheet. Then from google sheet, google form will be generated. This google form will send to google classroom. So student can give exam.

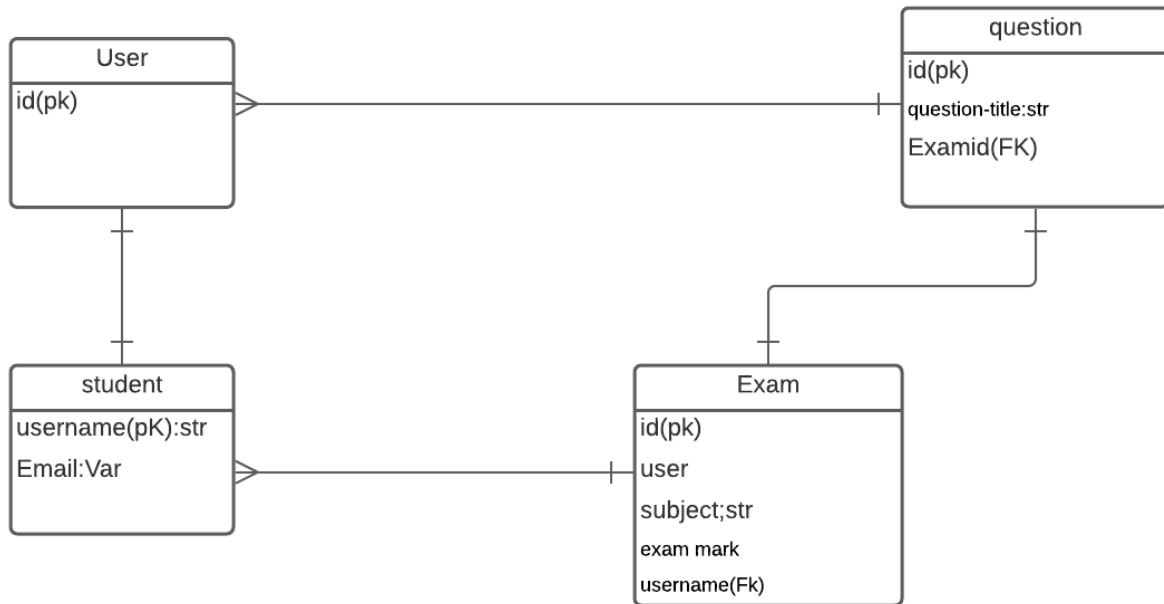
Class 8- Engineering

Class Diagram:

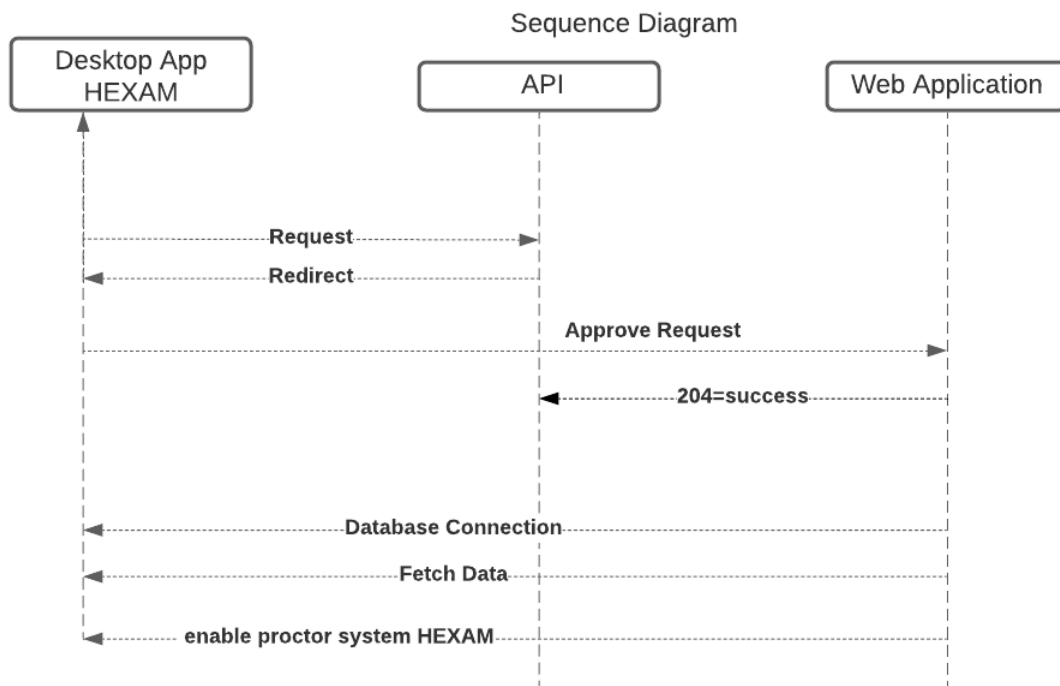


ERD Diagram:

ERD Diagram



Sequence Diagram:



Chapter 9- Development

Core modules coding samples:

```
requirements.txt × serializers.py × views.py × urls.py ×
1 from rest_framework import serializers
2 from .models import User, Exam, Question, Student
3
4 class UserSerializer(serializers.ModelSerializer):
5     class Meta:
6         model=User
7         fields=['id', 'user_name']
8
9 class ExamSerializer(serializers.ModelSerializer):
10    class Meta:
11        model=Exam
12        fields=['id', 'user', 'subject', 'exam_type', 'exam_mark', 'question', 'exam_date', 'duration']
13
14 class QuestionSerializer(serializers.ModelSerializer):
15    class Meta:
16        model=Question
17        fields=['id', 'question_title', 'option_first', 'option_second', 'option_third', 'option_fourth', 'mark']
18
```

```
        model=Exam
        fields=['id', 'user', 'subject', 'exam_type', 'exam_mark', 'question', 'exam_date', 'duration']

class QuestionSerializer(serializers.ModelSerializer):
    class Meta:
        model=Question
        fields=['id', 'question_title', 'option_first', 'option_second', 'option_third', 'option_fourth', 'mark']

class StudentSerializer(serializers.ModelSerializer):
    class Meta:
        model=Student
        fields=['user_name', 'email', 'image']
```

```

requirements.txt × serializers.py × views.py × urls.py ×
from django.shortcuts import render, redirect
from django.http import HttpResponse
from .forms import UserForm, SubjectForm, QuestionForm, AnswerForm, ExamForm
from .models import User, Subject, Question, Answer, Exam, Student
from random import randint
from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework.generics import ListAPIView, RetrieveAPIView
from .serializers import UserSerializer, ExamSerializer, QuestionSerializer, StudentSerializer
from rest_framework.viewsets import ModelViewSet
from rest_framework.parsers import FormParser, MultiPartParser

# Create your views here.
class UserAPIView(ModelViewSet):
    queryset=User.objects.all()
    serializer_class=UserSerializer

```

```

class UserAPIView(ModelViewSet):
    queryset=User.objects.all()
    serializer_class=UserSerializer

class ExamAPIView(ModelViewSet):
    queryset=Exam.objects.all()
    serializer_class=ExamSerializer

class QuestionAPIView(ModelViewSet):
    queryset=Question.objects.all()
    serializer_class=QuestionSerializer

class StudentAPIView(ModelViewSet):
    queryset = Student.objects.all()
    serializer_class = StudentSerializer
    parser_classes = [FormParser, MultiPartParser]

```

```

requirements.txt x serializers.py x views.py x urls.py x
1
2 import ...
9
10 router=DefaultRouter()
11 router.register(r'api/user',views.UserAPIView,basename='api_user')
12 router.register(r'api/exam',views.ExamAPIView,basename='api_exam')
13 router.register(r'api/question',views.QuestionAPIView,basename='api_question')
14 router.register(r'api/student',views.StudentAPIView,basename='api_student')
15
16
17 urlpatterns = [
18     path('admin/', admin.site.urls),
19     path('',views.userForm, name='userForm'),
20     path('subject/',views.subjectForm, name='subjectForm'),
21     path('question/',views.questionForm,name='questionForm'),
22     path('answer/', views.answerForm, name='answerForm'),
23     path('exam/',views.examForm,name='examForm'),
24     path('showsubject/',views.showSubject,name='showSubject'),

```

Apps Script Untitled project

```

Files +
Code.gs
Libraries +
Services +

1 // Replace the variables in this block with real values.
2 var address = 'database_IP_address';
3 var user = 'user_name';
4 var userPwd = 'user_password';
5 var db = 'database_name';
6
7 var dbUrl = 'jdbc:mysql://' + address + '/' + db;
8
9 // Read up to 1000 rows of data from the table and log them.
10 function readFromTable() {
11     var conn = Jdbc.getConnection(dbUrl, user, userPwd);
12
13     var start = new Date();
14     var stmt = conn.createStatement();
15     stmt.setMaxRows(1000);
16     var results = stmt.executeQuery('SELECT * FROM entries');
17     var numCols = results.getMetaData().getColumnCount();
18
19     while (results.next()) {
20         var rowString = '';
21         for (var col = 0; col < numCols; col++) {
22             rowString += results.getString(col + 1) + '\t';
23         }
24         Logger.log(rowString)
25     }
26
27     results.close();
28     stmt.close();
29
30     var end = new Date();
31     Logger.log('Time elapsed: %sms', end - start);
32 }

```

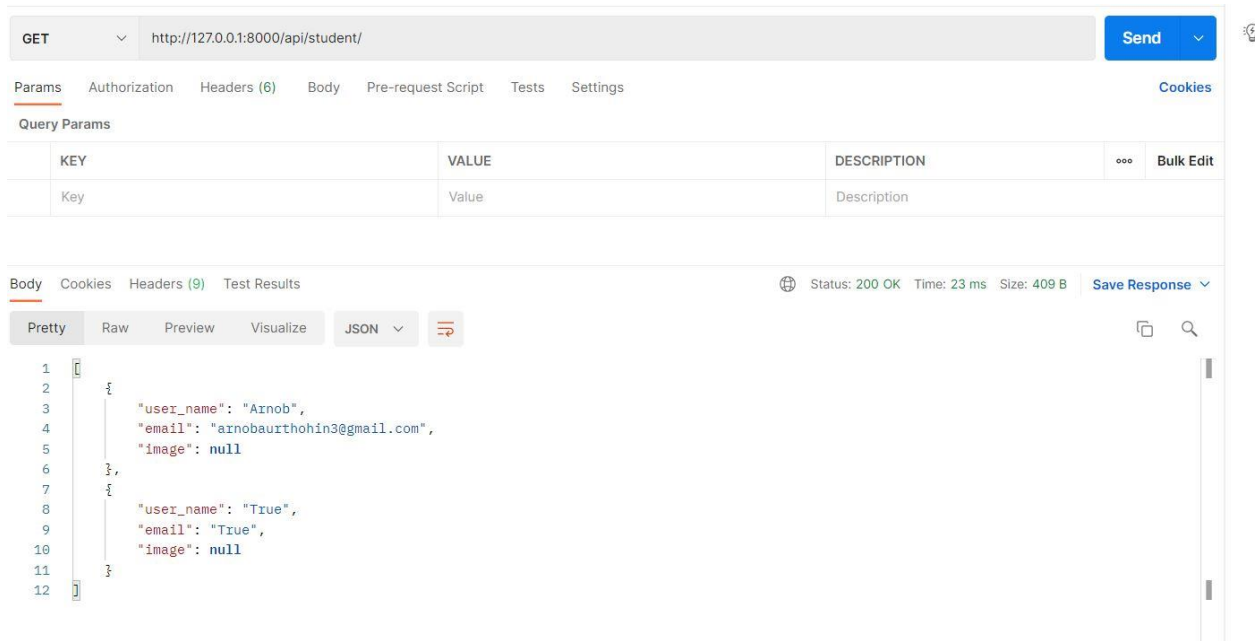
Possible problem breakdown: Many ways I can use for my system breakdown. Breakdown will make the system easier for development. Here is my breakdown solution is given below.

- Analyze the overall scenario.
- Analyze the development level.
- Design the database
- Analyze which API connection will be perfect.
- Backend development.
- Design the API.
- Develop the API.
- Check the API connection.
- Finalize the development.
- Connect the both web application and desktop app HEXAM.

This is the possible breakdown solution.

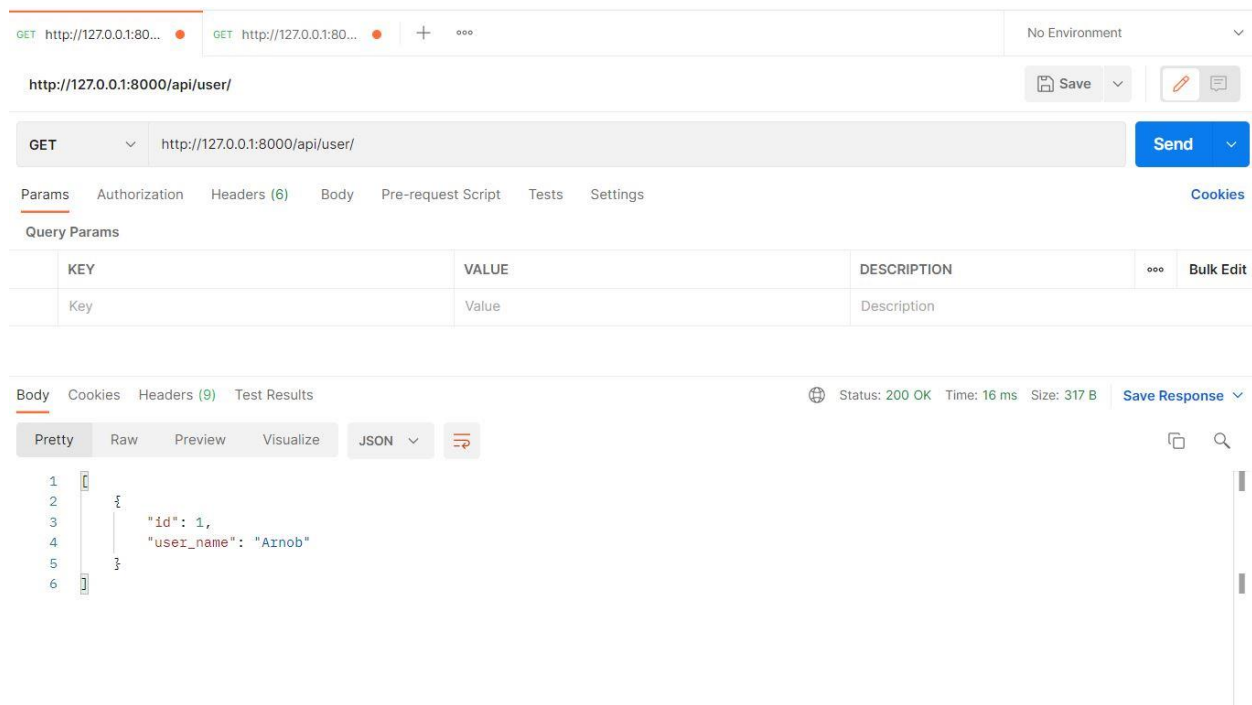
Chapter 10-Testing

Test Case Name	API Testing		
Test Class	student		
Test Description	Checking API 'student'		
Data Sources	Test Steps	Expected Result	Actual Result
	1)Test API by POSTMAN	1) Will show the data of student	1) Detail data of Student API



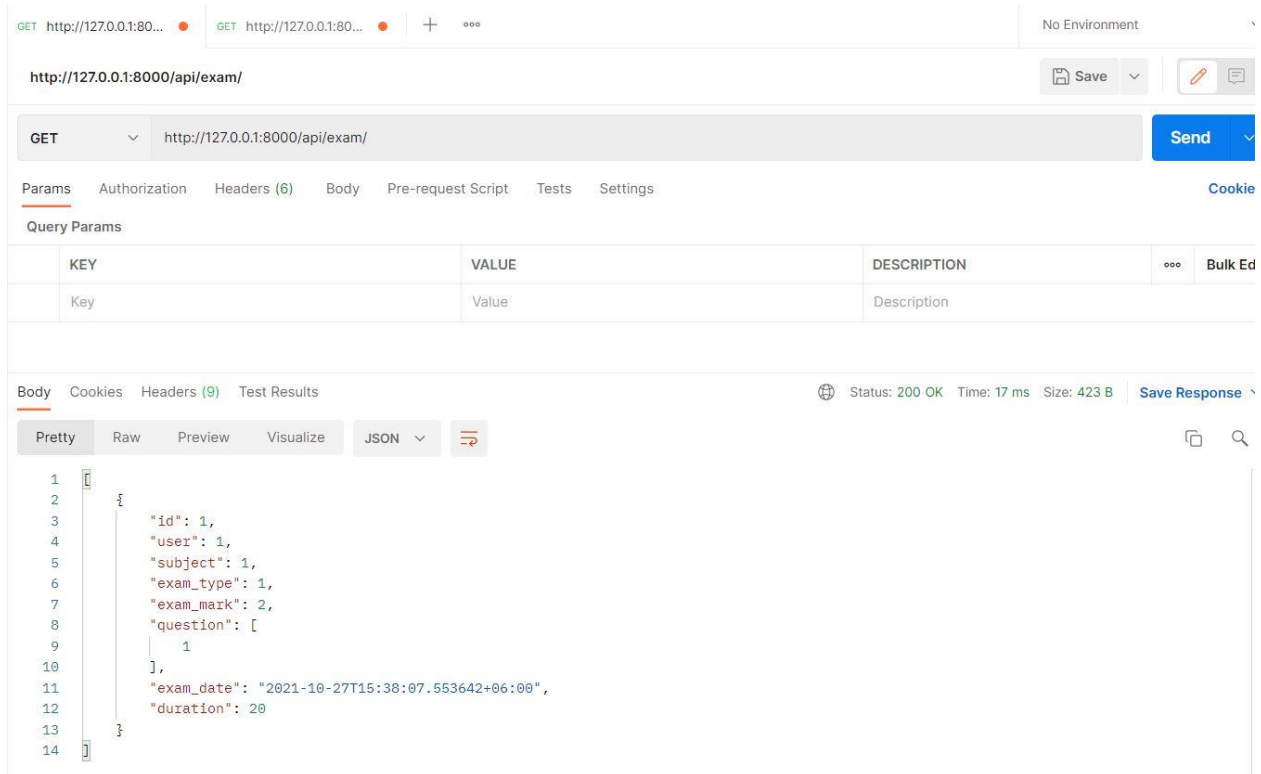
In here all the data of student is showing. So, API is connected successfully and will transfer data.

Test Case Name	API Testing		
Test Class	user		
Test Description	Checking API 'user'		
Data Sources	Test Steps	Expected Result	Actual Result
	1)Test API by POSTMAN	1) Will show the data of user	1) Detail data of user API



In here all the data of user is showing. So, API is connected successfully.

Test Case Name	API Testing		
Test Class	exam		
Test Description	Checking API 'exam'		
Data Sources	Test Steps	Expected Result	Actual Result
	1)Test API by POSTMAN	1) Will show the data of exam	1) Detail data of exam API



In here all the data of exam is showing. So, API is connected successfully.

Test Case Name	API Testing		
Test Class	question		
Test Description	Checking API 'question'		
Data Sources	Test Steps	Expected Result	Actual Result
	1)Test API by POSTMAN	1) Will show the data of question	1) Detail data of question API

The screenshot shows the Postman interface for a GET request to `http://127.0.0.1:8000/api/question/`. The request is successful with a status of 200 OK. The response body is displayed in JSON format, showing a question with ID 1, title "How many alphabets are in English?", four options, and a mark of 2.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```

1  {
2    "id": 1,
3    "question_title": "How many alphabets are in English?",
4    "option_first": "29",
5    "option_second": "24",
6    "option_third": "26",
7    "option_fourth": "34",
8    "mark": 2
9  }

```

In here all the data of question is showing. So, API is working successfully. All the testing is done successfully by POSTMAN. A software where we can test API.

Chapter 11- Implementation

Training: This type of system has introduced during this pandemic. When there is online class was happening but no system for online exam giving. So, we are working on a project name proctoring for online exam taking system. Here using this system for student and teacher need proper knowledge and computer skill. AI is enable here. And will also use webcam. Need proper instruction as a video and manual instruction also.

Chapter 12- Critical Appraisal and Evaluation

Success rate against each objectives: Success rate shows a project success. How much there is chance for success. The main system will be desktop app HEXAM. Everybody now days familiar with desktop application. In the desktop application there used pure python. Python is now most popular language. AI is enable. For API I have used rest API. All the functions works very smoothly and well managed. Function viewing is well set. So it has a great success rate.

How much better could have been done: This system name is mainly proctoring system. This project we are working is internship. So all the cant be prepare for this time. There are more efficient solution out there in the market. Need to keep updated with those system. Need to connect other system through API. And there are some requirement I put should have and could have, that I will work further.

How better the feature of the solution: This system has some best features. Here authority can control user and monitor. Random question generation. AI is enable where proctor mainly works. Question will come through the web application through API. Without electricity user can give exam. And there are some features we need to work on.

Which features could not be touched: Need to adjust 360 degree camera and also voice recognition and more AI needs to be enable. Need to connect more system through API.

Chapter 13- Conclusion

Summary of the project: This project name is Proctoring. Where one can give exam. In here there will be a main desktop application name HEXAM and a web based application. Where the random question will be generated. It will connect to the desktop app through API. API is used for data transfer. All kinds of AI is enabled to make the online exam like offline for monitoring. For better understanding all the charts and diagram have been showed.

Goal of the project: Every system want to achieve some certain goal for sure. My projects main goal is to take exam online and monitor the online exam.

My experience: My experience working in this project is good. This is our internship program. In here our work is divided. Doing my part is challenging. For doing my part I have taken help from different web sources. I continuous communicate with my supervisor. Doing this work I have learnt a lot that will help me in future.

Appendices

Test Scripts:

Test Case Name	API Testing		
Test Class	question		
Tester	Walid Hasan		
Test Description	Checking API 'question'		
Data Sources	Steps of testing	Expect Result	Actual Result
	1)Test API by POSTMAN	1) Will show the data of question	1) Detail data of question API

The screenshot displays the Postman interface for a GET request to `http://127.0.0.1:8000/api/question/`. The request is successful, returning a 200 OK status with a response time of 11 ms and a size of 443 B. The response body is shown in JSON format:

```
{
  "id": 1,
  "question_title": "How many alphabets are in English?",
  "option_first": "29",
  "option_second": "24",
  "option_third": "26",
  "option_fourth": "34",
  "mark": 2
}
```

Successfully working.

Test Case Name	API Testing		
Test Class	student		
Tester	Walid Hasan		
Test Description	Checking API 'student'		
Data Sources	Test Steps	Expected Result	Actual Result
	1)Test API by POSTMAN	1) Will show the data of student	1) Detail data of Student API

GET http://127.0.0.1:8000/api/student/ Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 23 ms Size: 409 B Save Response

Pretty Raw Preview Visualize JSON

```

1  [
2    {
3      "user_name": "Ainob",
4      "email": "ainobaurthohin3@gmail.com",
5      "image": null
6    },
7    {
8      "user_name": "True",
9      "email": "True",
10     "image": null
11   }
12 ]

```

Successfully working.

System code:

```

requirements.txt x serializers.py x views.py x urls.py x
1 from rest_framework import serializers
2 from .models import User, Exam, Question, Student
3
4 class UserSerializer(serializers.ModelSerializer):
5     class Meta:
6         model=User
7         fields=['id', 'user_name']
8
9 class ExamSerializer(serializers.ModelSerializer):
10    class Meta:
11        model=Exam
12        fields=['id', 'user', 'subject', 'exam_type', 'exam_mark', 'question', 'exam_date', 'duration']
13
14 class QuestionSerializer(serializers.ModelSerializer):
15    class Meta:
16        model=Question
17        fields=['id', 'question_title', 'option_first', 'option_second', 'option_third', 'option_fourth', 'mark']
18
19

```

```

        model=Exam
        fields=['id', 'user', 'subject', 'exam_type', 'exam_mark', 'question', 'exam_date', 'duration']

class QuestionSerializer(serializers.ModelSerializer):
    class Meta:
        model=Question
        fields=['id', 'question_title', 'option_first', 'option_second', 'option_third', 'option_fourth', 'mark']

class StudentSerializer(serializers.ModelSerializer):
    class Meta:
        model=Student
        fields=['user_name', 'email', 'image']

```

```

requirements.txt × serializers.py × views.py × urls.py ×
from django.shortcuts import render, redirect
from django.http import HttpResponse
from .forms import UserForm, SubjectForm, QuestionForm, AnswerForm, ExamForm
from .models import User, Subject, Question, Answer, Exam, Student
from random import randint
from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework.generics import ListAPIView, RetrieveAPIView
from .serializers import UserSerializer, ExamSerializer, QuestionSerializer, StudentSerializer
from rest_framework.viewsets import ModelViewSet
from rest_framework.parsers import FormParser, MultiPartParser

# Create your views here.
class UserAPIView(ModelViewSet):
    queryset=User.objects.all()
    serializer_class=UserSerializer

```

Apps Script Untitled project

Files	+	↶ ↷	▶ Run	⌘ Debug	readFromTable ▼	Execution log
Code.gs						
Libraries	+					
Services	+					

```

1 // Replace the variables in this block with real values.
2 var address = 'database_IP_address';
3 var user = 'user_name';
4 var userPwd = 'user_password';
5 var db = 'database_name';
6
7 var dbUrl = 'jdbc:mysql://' + address + '/' + db;
8
9 // Read up to 1000 rows of data from the table and log them.
10 function readFromTable() {
11     var conn = Jdbc.getConnection(dbUrl, user, userPwd);
12
13     var start = new Date();
14     var stmt = conn.createStatement();
15     stmt.setMaxRows(1000);
16     var results = stmt.executeQuery('SELECT * FROM entries');
17     var numCols = results.getMetaData().getColumnCount();
18
19     while (results.next()) {
20         var rowString = '';
21         for (var col = 0; col < numCols; col++) {
22             rowString += results.getString(col + 1) + '\t';
23         }
24         Logger.log(rowString)
25     }
26
27     results.close();
28     stmt.close();
29
30     var end = new Date();
31     Logger.log('Time elapsed: %sms', end - start);
32 }

```

References

Agile Software Development. (n.d.). Retrieved from https://www.google.com/search?q=agile+software+development&rlz=1C1CHBD_enBD895BD895&oq=Agile+&aqs=chrome.1.69i57j69i59l2j0i433i512j0i512l6.3773j0j7&sourceid=chrome&ie=UTF-8

ExamOnline. (n.d.). Retrieved from <https://www.softwaresuggest.com/examonline>

Non functional requirement. (n.d.). Retrieved from <https://www.guru99.com/non-functional-requirement-type-example.html>

Plagiarism Report

Turnitin Originality Report

Processed on: 20-Nov-2021 14:43 +06
 ID: 1708443764
 Word Count: 3424
 Submitted: 1

Similarity Index
9%

Similarity by Source

Internet Sources: 8%
 Publications: 1%
 Student Papers: 8%

183-16-349 By Md. Walid
 Hasan

4% match (student papers from 25-Jun-2020)

[Submitted to Daffodil International University on 2020-06-25](#)

3% match (Internet from 17-Nov-2021)

http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5796/191-16-437%20%20%2815_%29.pdf?isAllowed=y&sequence=1

1% match (student papers from 04-Apr-2018)

Class: Article 2018

Assignment: Journal Article

Paper ID: [940919861](#)

1% match (student papers from 30-Jul-2020)

[Submitted to Daffodil International University on 2020-07-30](#)

1% match (Internet from 17-Nov-2021)

http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/5766/182-16-344%20%288_%29.pdf?isAllowed=y&sequence=1